

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Ondřej David

Mikrosimulace městské silniční dopravy

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Petr Hnětynka, Ph.D.

Studijní program: Informatika, Programování

2007

Děkuji vedoucímu bakalářské práce RNDr. Petru Hnětynkovi, Ph.D. za podnětné rady a připomínky při vývoji projektu a vypracování této bakalářské práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 10. května 2007

Ondřej David

Obsah

1	Úvod	6
1.1	Simulace	6
1.2	Cíl projektu	7
1.3	Struktura práce	7
2	Obecný popis komponent simulace	8
2.1	Silniční síť	8
2.2	Silniční koridor	8
2.3	Jízdní pruh	8
2.4	Úsek jízdního pruhu	9
2.5	Křižovatka	9
2.6	Algoritmus propojení dvou jízdních pruhů	9
2.7	Dopravní informace	14
2.8	Vozidlo	14
2.9	Řidič	14
3	Struktura implementace	15
3.1	Úsek jízdního pruhu	15
3.2	Jízdní pruh	15
3.3	Silniční koridor	16
3.4	Křižovatka	17
3.5	Vozidlo	18
3.6	Řidič	19
3.7	Grafické rozhraní uživatele	19
3.8	Krok simulace	19
3.9	Vstup a výstup simulace	20
4	Umělá inteligence řidiče "driver1"	21
4.1	Stav řidiče	21
4.2	Zpracování událostí	22
4.3	Definice rychlosti volného toku	22
4.4	Určení rychlosti volného toku	22
4.5	Určení zrychlení a zpomalení	23
4.6	Bezpečná vzdálenost za vozidlem	23

4.7	Vliv vyhlídky na oblast s odlišnou rychlostí volného toku na stav řidiče	23
4.8	Vliv vozidla před řidičem na jeho stav	25
4.9	Predikční model v křižovatce	31
4.10	Zastavit před nebo v křižovatce?	31
4.11	Impulzy k změně jízdního pruhu	32
4.12	Aktualizace stavu	32
4.13	Další implementace	33
5	Shrnutí	34
5.1	Výsledná podoba projektu	34
5.2	Test simulátoru	34
5.3	Jak projekt rozšířit	35
6	Srovnání s projektem Simulace automobilového provozu	37
7	Závěr	38
	Literatura	39

Název práce: Mikrosimulace městské silniční dopravy
Autor: Ondřej David
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. Petr Hnětynka, Ph.D.
e-mail vedoucího: hnetynka@dsrg.ms.mff.cuni.cz

Abstrakt: Tato bakalářská práce se zabývá popisem programu pro modelování městské silniční sítě, jež je určena k mikrosimulaci dopravy. Součástí práce je také popis jednoho z algoritmů používaných pro simulaci. První část je zaměřena na popis vnitřní struktury programu, tzn. reprezentace silniční sítě a částí, z kterých se skládá, jaké je možné zadat parametry simulace a nakonec jak interpretovat naměřená data. Druhá část se zabývá jedním z algoritmů, které se pro simulaci dají použít. Je zde popsán způsob, jak virtuální řidič zpracovává události silničního provozu, jak si hlídá bezpečnou vzdálenost za vozidlem, který jede před ním, jeho strategii v křižovatce, na čem závisí jeho rozhodnutí, zda křižovatkou projede či nikoli a nakonec co ho vede k tomu přehradit se z pruhu do pruhu.

Klíčová slova: mikrosimulace, řidič, silnice, křižovatka

Title: Microsimulation of urban road traffic
Author: Ondřej David
Department: Department of Software Engineering
Supervisor: RNDr. Petr Hnětynka, Ph.D.
Supervisor's e-mail address: hnetynka@dsrg.ms.mff.cuni.cz

Abstract: This work deals with the description of the program for modelling urban road network designed for microsimulation of traffic. Part of the work is devoted to description of one of the algorithms available for use in simulation. The first part contains information about the internal structure of the program, which means the way of representation of the road network, the parts it is constituted of, what parameters of simulation are available to input and finally how to interpret data collected during simulation. The second part depicts one of the available simulation algorithms. It considers procession of traffic events, how the driver maintains safe distance behind the vehicle in front of him, his strategy in crossroads, what facts make him go and stop and eventually what are the reasons for changing lane.

Keywords: microsimulation, driver, road, crossroad

Kapitola 1

Úvod

1.1 Simulace

Simulační modelování je technika používaná k analýze chování složitých dynamických systémů, které není možné popsat analytickými prostředky. Systém se skládá z množství komponent, entit, mezi nimiž dochází k interakci.

Doprava se může podle [6] vnímat jako komplexní dynamický systém. Vytváření makromodelů je jeden z nejběžnějších přístupů k modelování takovýchto velkých celků. Makromodely se řídí přístupem "shora dolů" a zaměřují se pouze na viditelné chování systému. Toto chování se snaží nadefinovat a zpětně vytvářet na základě úhrnných parametrů. V případě dopravy jsou modely většinou odvozeny z dynamiky tekutin, kde úhrnné parametry představují předpokládaný objem dopravy a průměrnou rychlost na dopravních tepnách silniční sítě.

Simulace založené na makromodelech mají výhodu poměrně krátké doby výpočtu, kde hlavní zátěž výpočtu vychází pouze z úhrnných parametrů systému. Makroskopické modely jsou užitečné v případech, kdy postačí hrubý odhad prvků simulačního modelu jako je například integrovaný varovný systém. Avšak většina aspektů komplexních systémů je velice nelineární, a tedy jsou silně citlivé na počáteční podmínky. Dokonce minimální změna těchto podmínek může mít libovolně velký dopad na celkové chování modelu. Výsledek simulace je spíše hrubý odhad chování systému. Věrohodné výsledky jsou pouze za ideálních podmínek.

Alternativní přístup, který potenciálně dokáže dát věrohodnější výsledky simulace, je mikromodelování. Je to přístup "zdola nahoru", kde se na komplexní systém nahlíží jako na velkou množinu malých spolukomunikujících komponent. Hlavní zřetel se klade na jednotlivé komponenty, jejich lokální pozici a podmínky, na základě kterých se odvodí jejich chování. V případě dopravy se zřetel klade na chování řidičů, jejich jízdu za jiným vozidlem

nebo postup při změně jízdního pruhu.

Problém mikrosimulace jsou její výpočetní nároky a ceny vývoje softwaru. Mikrosimulace běží na velmi detailní úrovni s tím, že se musí zpracovat chování každé entity systému. Tím pádem dochází k velmi intenzivním výpočtům a nárokům na rychlost procesoru.

1.2 Cíl projektu

Cílem projektu je vytvořit program, který bude umožňovat vytvořit si vlastní silniční síť, kterou bude možné parametrizovat očekávanou hustotou dopravy, očekávaným průtokem provozu křižovatkou v jednotlivých směrech a dalšími vlastnostmi. Tuto síť bude možné testovat na zvoleném modulu chování řidiče za účelem naměřit statistická data, která budou vypovídat o schopnosti vozovky poskytnout potřebné kapacity.

1.3 Struktura práce

Následující kapitola popisuje obecné vlastnosti komponent programu a jejich provázanost. Třetí kapitola blíže rozebírá stavbu jednotlivých částí programu a jaké každá z nich plní úkoly. Čtvrtá kapitola nabízí pohled do implementace modulu řidiče. Pátá kapitola shrnuje celkovou podobu projektu a nastiňuje potenciál jeho dalšího rozšíření. Šestá kapitola srovnává projekt s jinou prací, která řeší podobnou problematiku. Závěr pak obsahuje rekapitulaci projektu a jeho využití.

Kapitola 2

Obecný popis komponent simulace

Program se skládá z části simulační, jejíž hlavní činností je obsluha simulace, a z části prezentační, která představuje grafické rozhraní pro uživatele. Kapitola se věnuje popisu komponent simulační části z hlediska jejich role v programu a obecného fungování. Jejich rozčlenění se opírá o navržené podobné struktury popsané v [4].

2.1 Silniční síť

Silniční síť se skládá z křižovatek, které jsou mezi sebou propojené silničními koridory. Konce koridorů, které nevedou do křižovatky, jsou místa, kde se do sítě generují nová vozidla a odkud vozidla ze sítě vyjíždí.

2.2 Silniční koridor

V Každém silničním koridoru se provoz může ubírat jedním nebo dvěma opačnými směry. Skládají se z navzájem rovnoběžných jízdnic pruhů. Pro každý jízdnic pruh platí, že jeho pravý sousední pruh nemůže být opačného směru než je pruh samotný. Počet jízdnic pruhů se může v koridoru a jeho jednotlivých směrech měnit. Pokud jeden ze směrů koridoru ústí do křižovatky, má informace o podílu toku provozu do každého jednotlivého směru, kterým lze z něj na křižovatce jet. Vozidla jezdí v nastaveném směru silničního koridoru.

2.3 Jízdnic pruh

Jízdnic pruh se skládá z tzv. úseků jízdnic pruhu. Vozidla jedoucí v jeho oblasti se pohybují pouze po jeho středu a nemají možnost manévrovat ze strany na stranu. Pro vozidla a jejich řidiče jsou v pruhu nachystány dopravní

informace, více v sekci 2.7. Pokud jízdní pruh ústí do křižovatky, obsahuje informace o směrech, do kterých lze z něj na křižovatce jet.

2.4 Úsek jízdního pruhu

Úsek jízdního pruhu je základní stavební kámen jízdního pruhu. Jeho oblast pokrývá obdélník, jejímž středem prochází úsečka představující kolejnici, po které jezdí vozidla. Je provázán s úsekem, na který navazuje a který navazuje na něj. Takto propojená posloupnost tvoří jízdní pruh.

2.5 Křižovatka

Křižovatka je tvořena třemi nebo čtyřmi vstupními silničními koridory. Rozvržení přednosti v jízdě je implicitně nastaveno na přednost vozidel jedoucích zprava a protijedoucích vozidel. Explicitně lze nastavit která dvojice vstupních koridorů bude mít přednost v jízdě.

Křižovatka má na starost propojení jednotlivých jízdních pruhů respektive koridorů ústících do křižovatky. Propojení se provádí následujícím způsobem. Zvolí se silniční koridor a směr (doleva, rovně, doprava), kterým je třeba vytvořit propojení. Najde se silniční koridor, který se v tomto směru od zvoleného koridoru nachází. Poté zbývá vzít množinu jízdních pruhů, které vedou do křižovatky zvoleného silničního koridoru a lze z nich jet do zvoleného směru, a propojit je s jízdními pruhy vedoucích z křižovatky výše nalezeného silničního koridoru.

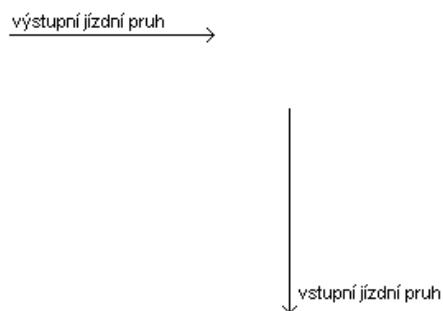
Postup propojení pruhů je zprava doleva. Pokud je více vstupních než výstupních jízdních pruhů, zbývající se propojí s nejlevějším výstupním jízdním pruhem.

2.6 Algoritmus propojení dvou jízdních pruhů

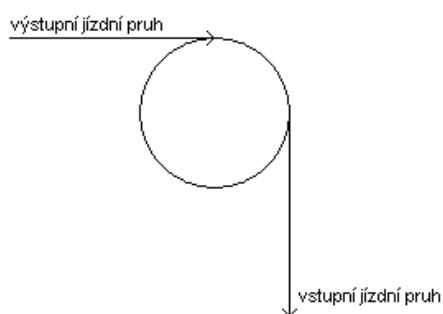
Algoritmus rozlišuje tři různé konfigurace pozice výstupního a vstupního jízdního pruhu.

1. V prvním případě (obr. č. 2.1) existuje kružnice, k níž je poslední úsek výstupního jízdního pruhu tečnou v jeho koncovém bodě a podobně je k ní první úsek vstupního jízdního pruhu tečnou v jeho počátečním bodě (obr. č. 2.2). Pokud se po kružnici vydáme z výstupního pruhu ve směru, kterým se úsek na kružnici napojuje, dorazíme k výstupnímu

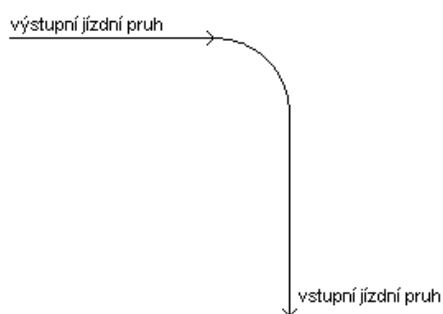
pruhu ze směru, z kterého se z kružnice odpojuje. Výsledný pruh je pak na obr. č. 2.3.



Obrázek 2.1: Vstupní pozice jízdních pruhů konfigurace č. 1.



Obrázek 2.2: Propojení pomocí kružnice.

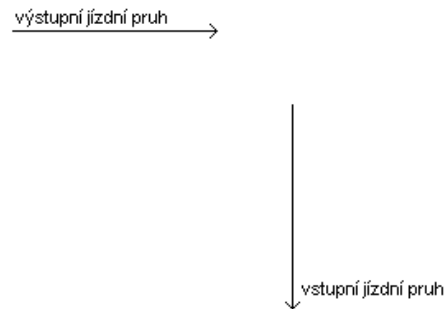


Obrázek 2.3: Výsledný propojený jízdní pruh.

2. Druhá konfigurace (Obr. č. 2.4) vychází z toho, že buď vstupní nebo výstupní jízdní pruh se dá prodloužit tak, aby se problém převedl na konfiguraci č. 1 (Obr. č. 2.5).

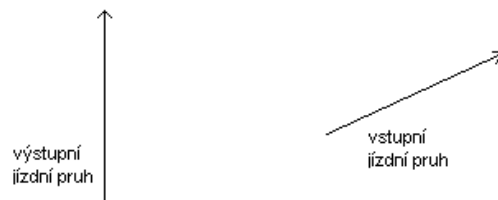


Obrázek 2.4: Vstupní pozice jízdních pruhů konfigurace č. 2.

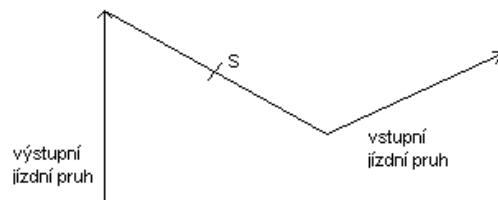


Obrázek 2.5: Prodloužení vstupního jízdního pruhu.

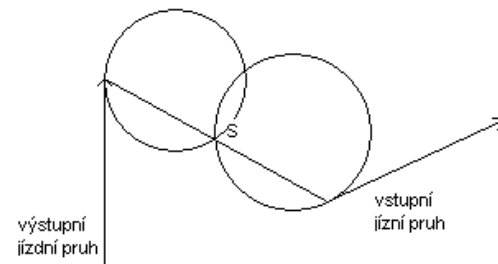
3. Třetí konfigurace (Obr. č. 2.6) je doplňkem konfigurace č. 2. Řešení spočívá v převedení situace na konfiguraci č. 2. Nejprve se najde střed S úsečky definované koncovým bodem výstupního jízdního pruhu a počátečním bodem vstupního jízdního pruhu (Obr. č. 2.7). Vytvoří se kružnice procházející bodem S , k níž je tečna poslední úsek výstupního jízdního pruhu v jeho koncovém bodě. Podobně se vytvoří kružnice ze strany vstupního jízdního pruhu (Obr. č. 2.8). Následuje vytvoření tečen t_1 a t_2 k právě vytvořeným kružnicím v bodě S (Obr. č. 2.9). Nyní zbývá vytvořit nový jízdní pruh (Obr. 2.11, 2.12) procházející bodem S , jehož směr bude definován směrnici osy tečen (Obr. č. 2.10) t_1 a t_2 tak, aby konfigurace výstupního a nového pruhu vyhovovala konfiguraci č. 2 stejně jako konfigurace nového a vstupního jízdního pruhu. Pruhy vytvořené pomocí algoritmu popsáném v konfiguraci č. 2 se pak spojí a vytvoří spojnici mezi výstupním a vstupním jízdním pruhem.



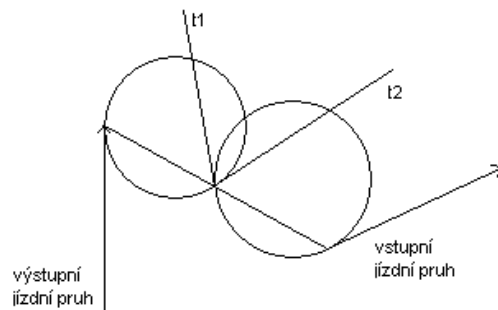
Obrázek 2.6: Vstupní pozice jízdních pruhů konfigurace č. 3.



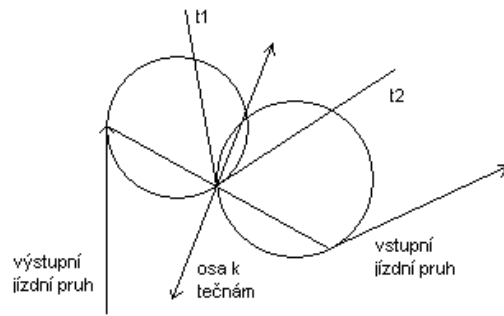
Obrázek 2.7: Střed úsečky propojující jízdní pruhy.



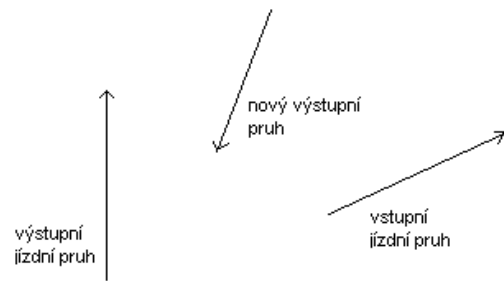
Obrázek 2.8: Pomocné kružnice.



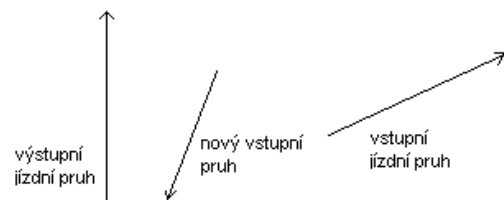
Obrázek 2.9: Tečny ke kružnicím procházející bodem S.



Obrázek 2.10: Osa k tečnám t_1 , t_2 .



Obrázek 2.11: Nový výstupní jízdní pruh.



Obrázek 2.12: Nový vstupní jízdní pruh.

2.7 Dopravní informace

Dopravní informace poskytují vozidlům a jejich řidičům informace týkající se vlastností buď celé vozovky nebo pouze konkrétního jízdního pruhu, ve kterém se právě nacházejí. Informace jsou uloženy v jednotlivých jízdních pruzích a jejich pozice je definovaná vzdáleností od začátku tohoto pruhu. Následuje výčet několika nejdůležitějších dopravních informací:

”**Směry jízdních pruhů před křižovatkou**” informuje o počtu jízdních pruhů před křižovatkou a jakými směry se z nich dá křižovatkou jet.

”**Hranice křižovatky**” informuje o rozhraní křižovatky a silničního koridoru.

”**Jízdní pruhy v křižovatce**” informuje o jízdních pruzích v křižovatce, ze kterých si řidič může vybrat kterým pojedje.

”**Průsečíky s jízdními pruhy**” informuje o množině průsečíků jízdního pruhu, ve kterém se informace nachází, s ostatními jízdními pruhy v křižovatce.

”**Zatáčka**” informuje o zatáčce a jejím poloměru.

”**Konec zatáčky**” informuje o konci zatáčky.

”**Změna počtu jízdních pruhů**” informuje o změně počtu jízdních pruhů jízdního koridoru, v kterém se informace nachází.

2.8 Vozidlo

Vozidla jsou předmětem ovládní řidičů. Jejich technické parametry, mezi které patří např. součinitel smykového tření se suchým asfaltem, mají pouze informační charakter pro řidiče, kteří mají možnost si volit rychlost a zrychlení libovolně. Během simulace jejich primární úkol je sbírat dopravní informace, aktualizovat kontext okolního prostředí a o změnách pravidelně informovat řidiče.

2.9 Řidič

Řidiči reprezentují umělou inteligenci, která řídí vozidla. Jsou schopni přijímat zprávy dopravního provozu a informace týkající se změn prostředí, ve kterém se jejich vozidlo nachází.

Kapitola 3

Struktura implementace

3.1 Úsek jízdního pruhu

Úsek jízdního pruhu je definován dvěma body v rovině, mezi kterými se vytvoří úsečka reprezentující kolejnici jízdního pruhu. Tato úsečka prochází středem obdelníku, který pokrývá oblast úseku a jehož šířka je 3.25m. Šířka byla zvolena na základě doporučení v návrhu silniční sítě v Olomouckém kraji popsaném v [7].

Úseky se dělí na dva druhy podle toho, jestli jsou součástí zatáčky či nikoli. Pokud je úsek součástí zatáčky, obsahuje v sobě dodatečné informace ohledně pozice středu kružnice, na jejíž části je zatáčka definovaná. Úsek má s obvodem kružnice společný svůj koncový a počáteční bod. Na začátku každé posloupnosti úseků pruhu v zatáčce je dopravní informace referující o zatáčce. Přímka procházející středem kružnice zatáčky a počátečním bodem úseku zatáčky svírá s přímkou procházející středem kružnice zatáčky a koncovým bodem úseku úhel 15 stupňů.

3.2 Jízdní pruh

Jak již bylo zmíněno v předešlé kapitole, jízdní pruh tvoří posloupnost vzájemně propojených úseků jízdního pruhu. Každý úsek je očíslovaný svým pořadovým číslem v posloupnosti. Očíslování je vzestupné, nebo sestupné. Podle toho se určuje směr jízdního pruhu v rámci jednoho silničního koridoru.

Jízdní pruh má na starost správu nad vozidly, které se v něm pohybují. Ukládá si jejich reference ve stromové struktuře setříděné podle jejich vzdálenosti od začátku tohoto pruhu za účelem snadného vyhledávání relativní pozice mezi vozidly.

Jízdní pruh v sobě obsahuje dopravní informace určené pro vozidla, která

jím projíždějí. Informace jsou uloženy setříděně ve spojovém seznamu podle vzdálenosti jejich pozice od začátku tohoto pruhu. Vozidlo si při každém zařazení do pruhu od něj vyžádá iterátor na první dopravní informaci, která se před ním nachází.

Pozice jízdního pruhu v silničním koridoru je definována jeho pořadím v posloupnosti sousedících pruhů stejného směru. Pořadí začíná na hodnotě 0 a je vzestupné zleva doprava.

Pokud je pruh obsažen v silničním koridoru, který je svázán s jinými silničními koridory, jízdní pruh je provázán s jízdními pruhy těchto koridorů, jejichž pořadí je stejné jako pořadí samotného pruhu.

Pokud jízdní pruh ústí do křižovatky, informace ohledně množiny směrů, do kterých lze z něj jet, je uložena v datovém typu integer, kde každý směr je reprezentován bitem. Pokud bit obsahuje 1, lze tímto směrem z pruhu do křižovatky jet. V opačném případě obsahuje 0.

3.3 Silniční koridor

Silniční koridor obsahuje pro oba směry spojový seznam jízdních pruhů. Pruhy jsou v seznamech setříděny podle jejich pořadí od středu vozovky směrem doprava.

Pokud se některý směr skládá pouze z jízdních pruhů, které na žádné jiné pruhy nenavazují, silniční koridor má za úkol do těchto pruhů generovat vozidla. Aby se docílilo toho, že se vozidla nebudou generovat v přesných intervalech za sebou, vytvoříme si časový interval, v kterém náhodně vyberem přesný časový bod, v němž vozidlo vygenerujeme a zajistíme, aby se v tomto intervalu už žádné jiné vozidlo nevygenerovalo. Hodnota

$$t_{interval} = \frac{60.0}{n};$$

označuje délku intervalu, v rámci kterého se může vygenerovat vozidlo, kde n je počet vygenerovaných vozidel za minutu. Hodnota

$$t = \text{random_generator.nextDouble}(t_{interval} - t_m);$$

pak udává přesný časový bod v tomto intervalu, v kterém se má vozidlo vygenerovat, kde t_m je minimální doba, která musí uplynout mezi generováním dvou vozidel.

Pokud vozovka vede do křižovatky, informace o podílu toku provozu do jednotlivých směrů jsou uloženy v poli reprezentující distribuční funkci náhodné veličiny podílů toku provozu. Při výběru náhodného směru v křižovatce se

použije kvantilová funkce, která najde index v tomto poli a tím i směr, který je s tímto prvkem v poli spojený.

Během simulace má vozovka za úkol přenášet informace mezi řidiči. Např. pokud nějaké vozidlo zapne blinkr, vozovka se postará o to, aby se informace o události dostala k řidiči, pro kterého je událost relevantní.

3.4 Křižovatka

Křižovatka plní svůj hlavní úkol před začátkem simulace. Vytvoří jízdni pruhy propojující silniční koridory a uloží si je do stromové mapy zobrazující množinu trojic (zdrojový silniční koridor, zdrojový jízdni pruh, směr) na množinu seznamů jízdni pruhů v křižovatce vycházejících ze silničního koridoru a jízdni pruhu ve směru uložených v trojici. Poté následuje vložení nezbytných dopravních informací do vytvořených jízdni pruhů.

Aby řidiči věděli o místech, kde se jejich dráha v křižovatce kříží s dráhou jiných vozidel, je třeba označit místa, kde se nachází průsečíky jejich jízdni pruhu v křižovatce s ostatními jízdni pruhy a vytyčit jejich délky.

Algoritmus nalezení vzdálenosti k začátku průsečíku dvou jízdni pruhů:

Kolem každého úseku obou jízdni pruhů se vytvoří ohraničení vymezující jeho oblast. Hranice je po stranách určena dvěma s kolejnici rovnoběžnými přímkami, od které má každá vzdálenost rovnou polovině šířky jízdni pruhu. Horní resp. dolní hranice je osa kolejnicí daného úseku a jeho následujícího resp. předcházejícího úseku. Jakmile je k dispozici toto ohraničení, je triviální určit, zda průnik oblastí dvou odlišných úseků pruhu je prázdný či nikoli a popřípadě pozici, kde k průniku dochází. Následující algoritmus demonstruje nalezení vzdálenosti k průsečíku od začátku jízdni pruhu.

```
distance := ∞;
for each usek1 ∈ pruh1 do
  for each usek2 ∈ pruh2 do
    begin
      intersection := getIntersection(usek1, usek2);
      if(intersection != NULL)
        begin
          auxDistance := pruh1.getDistanceTo(usek1) + intersection.distance;
          if(auxDistance < distance)
            distance := auxDistance;
        end
      end
    end
```

return distance;

Pokud `pruh1` a `pruh2` nevedou do stejného jízdního pruhu, po nalezení vzdálenosti k průsečíku se pruh obrátí a aplikuje se stejný algoritmus. Tím získáme délku průsečíku. Tímto způsobem se pro každý jízdní pruh v křižovatce nalezne množina všech průsečíků a jejich délky s ostatními pruhy. Nalezená množina se následně v jízdním pruhu uloží do dopravní informace.

V následující fázi je třeba označit místa, v kterých se rozcházejí jízdní pruhy vycházející ze stejného zdrojového pruhu. Využije se k tomu stejný algoritmus jako v předchozím odstavci. Vezme se dvojice takových pruhů, jeden z nich se obrátí a nalezne se průsečík. Rozdíl délky obráceného pruhu a vzdálenosti k průsečíku je vzdálenost od začátku tohoto pruhu k místu, kde se rozchází s druhým pruhem z dvojice. Do místa se ohledně tohoto vloží dopravní informace pro vozidlo, které bude mít za úkol informovat řidiče za sebou, že opustilo jeho jízdní pruh v křižovatce.

V konečné fázi se na konec každého jízdního pruhu ústícího do křižovatky umístí dopravní informace ohledně počtu těchto jízdních pruhů a do jakého směru lze z každého z nich na křižovatce jet.

3.5 Vozidlo

Mezi technické parametry vozidla patří jeho šířka, délka, hmotnost a součinitel smykového tření se suchým asfaltem k tomu, aby se dala vypočítat jeho maximální rychlost v zatáčce o daném poloměru.

Při kontrole dopravních informací se vozidlo dotáže řidiče na viditelnou vzdálenost. V rozsahu této vzdálenosti zkontroluje nové dopravní informace, informuje o nich řidiče a uloží si je do spojového seznamu a spolu s ní i vzdálenost, kterou bude muset urazit, než se k pozici informace dostane. Pamatovat si pozici informace je pro vozidlo důležité, protože pokud např. informuje o tom, že v její pozici dochází ke slévání jízdních pruhů, vozidlo bude mít na starost při průjezdu touto oblastí informovat řidiče v tomto pruhu o tom, že se do jejich pruhu zařadilo nové vozidlo.

Vozidlo má také na starost aktualizaci kontextu vozidla, z kterého řidič čerpá informace o svém prostředí. Obsahuje spojové seznamy dopravních informací stejného typu a ke každé z nich také vzdálenost, jakou musí vozidlo urazit, než se k ní dostane. Kontext tedy poskytuje informace o tom, jestli je vozidlo v zatáčce, poloměr této zatáčky, kdy končí nebo začíná nová zatáčka, jak daleko je k nejbližší křižovatce, kterým jízdním pruhům a jejich vozidlům je třeba dát přednost a další informace stejného charakteru.

Řidič má možnost ovládat vozidlo přes několik standardních funkcí. Patří

mezi ně zadání rychlosti a zrychlení, zapnutí/vypnutí blinkru nebo impuls k přeřazení do sousedního jízdního pruhu pod daným úhlem. Vozidlo se pak stará o předání příslušných událostí vozidlům, pro které jsou tyto události relevantní.

3.6 Řidič

Implementace řidiče reprezentuje jeho inteligenci a míru schopnosti řešit situace v silničním provozu. Takovou implementaci budeme nazývat modul chování řidiče. Všechny moduly mají společné abstraktní funkce, které musí implementovat, aby moduly mohly být použity k simulaci. Funkce především slouží ke komunikaci vozidla s řidičem. Např. v situaci kdy vozidlo kontroluje nové dopravní informace, potřebuje se řidiče zeptat jak daleko vidí, aby věděl do jaké vzdálenosti má kontrolu provést. Jakmile je k dispozici nová informace, vozidlo potřebuje o ní předat informace řidiči. Podobně řidič musí být schopný přijmout událost silničního provozu. Pokud je před vozidlem řidiče křižovatka, vozidlo se ptá řidiče kterým směrem se má v křižovatce vydat, aby podle toho mohl dostatečně v předstihu aktualizovat kontext prostředí. Nakonec hlavní smyčka programu má u řidiče k dispozici funkci, přes kterou mu oznámí, že má zareagovat na dopravní situaci. Tato funkce se volá v každém kroku simulace.

V projektu jsou naimplementovány tři moduly chování řidiče. Jeden z nich je popsán v kapitole 4. Zbylé dva jsou od něj odvozené.

3.7 Grafické rozhraní uživatele

Jádrem grafického rozhraní uživatele je interaktivní mapa, která je reprezentována mřížkou, do které uživatel zasazuje prvky silniční sítě. Mřížka je tvořena čtverečky, do nichž jsou uloženy bližší informace o objektech, jejichž oblast do čtverečku zasahuje. S její pomocí lze snáze najít objekt, nad kterým se nachází kurzor uživatele. Podle souřadnic kurzoru se určí čtvereček mřížky, nad jehož oblastí se kurzor nachází, a podle informací v něm uložených se rychle najde objekt, který chce uživatel ovládat.

3.8 Krok simulace

Při každém kroku simulace si nejdříve silniční koridory aktualizují statistická data o provozu a zkontrolují si, zda nemají vygenerovat do modelu nová vozidla. Dále všichni řidiči zareagují na aktuální situaci v silničním provozu a nakonec se vozidla posunou o vzdálenost podle zadané rychlosti a zrychlení. Krok trvá 50ms, aby se průběh simulace mohl plynule vykreslovat.

3.9 Vstup a výstup simulace

Simulační model, jako vstup aplikace, je uložen v souboru, kde je silniční síť spolu s parametry uložena strukturovaně ve formátu xml. Příručka k ovládání grafického rozhraní, ve kterém lze interaktivně model vytvořit, je obsažena v příloženém CD.

Výstup simulace je grafický a textový. Grafický výstup představuje pravidelné vykreslování průběhu simulace, v kterém je patrná propustnost silniční sítě. Textový výstup obsahuje statistická data jednotlivých silničních koridorů pro zvolený modul chování řidiče, která mohou být zobrazena v okně uživatele nebo se mohou uložit spolu se simulačním modelem do souboru. Statistická data zahrnují:

- Hodnota **průměrná rychlost** označuje průměrnou rychlost vozidel v koridoru od začátku simulace.
- Hodnota **aktuální průměrná rychlost** označuje aktuální průměrnou rychlost vozidel v koridoru.
- Hodnota **průměrná doba průjezdu** označuje průměrnou dobu, kterou trvá vozidlům projet koridor.
- Hodnota **nejkratší doba průjezdu** označuje nejkratší čas, v kterém se některému vozidlu simulace podařilo koridorem projet.
- Hodnota **nejdelší doba průjezdu** označuje nejdelší čas, který trval některému vozidlu simulace koridorem projet.
- Hodnota **tok vozidel** označuje počet vozidel, který projede koridorem za minutu.
- Hodnota **ucpáno** indikuje stav koridoru jestli je ucpaný nebo ne.

Kapitola 4

Umělá inteligence řidiče ”driver1”

Řidič je implementován jako konečný automat. Kapitola obsahuje informace týkající se množiny proměnných, jejichž hodnoty definují stav automatu. Je zde popsán způsob, jak řidič zpracovává informace a události silničního provozu, které vedou ke změně tohoto stavu. Dále je popsána strategie jak si udržet bezpečnou vzdálenost za jiným vozidlem, jak se rozhodovat při průjezdu křižovatkou a nakonec jaké jsou impulsy k změně jízdního pruhu.

4.1 Stav řidiče

Stav řidiče je složen z několika proměnných a indikátorů. Každý stav obsahuje časové kvantum, kterým je jeho platnost omezená a po jehož vypršení je třeba stav řidiče aktualizovat, pokud dřív nepřijde impuls k aktualizaci z dopravního provozu. Impuls může znamenat příchod buď dopravní události nebo nové dopravní informace viditelné z pozice řidiče.

Mezi další proměnné patří aktuální rychlost a zrychlení, počáteční, resp. konečná rychlost, která reprezentuje rychlost v době kdy se řidič do stavu přepl, resp. rychlost, která poslouží jako počáteční rychlost pro následující stav, pokud k aktualizaci nedojde před vypršení časového kvanta. Tyto hodnoty jsou přítomny hlavně z toho důvodu, že simulace probíhá v diskrétních krocích a hodnota aktuální rychlosti nemusí odpovídat rychlosti, do které se řidič chtěl dostat. Počáteční rychlost se ukládá z toho důvodu, aby řidič věděl jakou rychlost má mít po uplynutí daného časového kvanta. Ze stejného důvodu je také obsažena doba, jakou už řidič ve stavu strávil.

Nedílnou součástí stavu je indikátor toho, jestli se řidič nachází v aktuálním stavu kvůli tomu, že ho omezuje vozidlo před ním. Tato informace hraje roli v případech, kdy například řidič obdrží událost, že vozidlo před ním zrychluje. Pokud je indikátor nastaven na *true*, je třeba zvážit, jestli není třeba

aktualizovat stav. Pokud je indikátor nastaven na *false*, tato událost je pro řidiče nepodstatná. Díky tomu se předchází zbytečné aktualizaci stavu.

Další indikátory informují o tom, zda řidič nechává z levého nebo pravého sousedního jízdního pruhu jiné vozidlo se před něj přeřadit či nikoli. Informace je důležitá v případech, kdy řidič obdrží dopravní událost o tom, že vozidlo nalevo či napravo od něj vyplo blinkr. Pokud jsou indikátory nastaveny na *true*, je třeba zvážit aktualizaci stavu.

Pokud se řidič snaží změnit pruh, je třeba si pamatovat jeho důvod. Vezměme případ kdy řidič chce předjet vozidlo před ním pro jeho pomalou jízdu. Pokud vozidlo začne zrychlovat, je třeba tento úmysl přehodnotit.

4.2 Zpracování událostí

Podle [2] je reakční doba řidičů zhruba 1 sekunda. Řidič si proto událost vloží do seznamu událostí seřazeného podle okamžiku jejich příchodu a ke každé si pamatuje dobu, která musí uplynout do jejího zpracování. Tato doba se aktualizuje v každém kroku simulace.

4.3 Definice rychlosti volného toku

Rychlost volného toku je konstantní rychlost, kterou si řidič zvolí pro půjezd úseku vozovky s danými parametry aniž by byl omezen jinými vozidly. V dalším textu se bude pojem vyskytovat pod zkratkou RVT.

4.4 Určení rychlosti volného toku

Jak již napovídá definice, k určení této rychlosti je třeba znát parametry vozovky.

Patří k nim maximální povolená rychlost, která je implicitně nastavena na 50 km/h. Pokud by byla zvolena jiná maximální rychlost, její hodnota bude k dispozici z kontextu vozidla. Řidič obsahuje koeficient překročení rychlosti, který reprezentuje jeho disciplinovanost vůči předpisům. Při jeho vynásobení maximální povolenou rychlostí dostaneme rychlost volného toku. Hodnota tohoto koeficientu je nastavena na hodnotu 1.5, čili při maximální povolené rychlosti 50 km/h hodnota rychlosti volného toku bude 75 km/h.

V případě že se vozidlo nachází v zatáčce, je třeba brát v úvahu poloměr této zatáčky. Díky znalosti hmotnosti vozidla a koeficientu smykového tření pneumatik se suchým asfaltem, je možné vypočítat maximální rychlost, kterou vozidlo může jet v zatáčce o daném poloměru.

$$v_{max} = \sqrt{rgf}$$

r ... poloměr zatáčky
 g ... gravitační zrychlení
 f ... součinitel smykového tření

Řidič obsahuje koeficient přiměřené odstředivé síly, který se pohybuje v intervalu (0,1). Při jeho vynásobení s rychlostí v_{max} dostaneme velikost rychlosti volného toku v této zatáčce.

Výsledná rychlost volného toku se vypočítá jako minimum všech jednotlivých rychlostí tohoto charakteru vypočítaných na základě jednotlivých parametrů vozovky.

4.5 Určení zrychlení a zpomalení

Na základě statistických měření podle [1] má řidič hodnotu průměrného zrychlení a zpomalení z dané počáteční a koncové rychlosti pevně dáno. Zrychlení byla měřena z rychlosti 0 km/h na každý násobek rychlosti 10 km/h až do rychlosti 100 km/h. Podobně zpomalení byla měřena z každého násobku rychlosti 10 km/h od rychlosti 100 km/h na rychlost 0 km/h. Pokud chce řidič např. zrychlení z rychlosti 35 km/h na rychlost 70 km/h, vezme si zrychlení z rychlosti 0 na (70-35) km/h zaokrouhloveno vždy na nejvyšší nižší násobek rychlosti 10 km/h.

4.6 Bezpečná vzdálenost za vozidlem

Bezpečná vzdálenost, kterou si řidič udržuje za vozidlem, je především závislá na jeho rychlosti. Vzdálenost se v tomto případě nebude měřit v metrech, ale v sekundách. BESIP doporučuje udržovat vzdálenost za vozidly 2 s především na dálnicích a rychlostních silnicích. Řidiči mají nastavenou tuto hodnotu na dobu 1.5 s. Vzdálenost mezi vozidly při rychlosti 72 km/h pak bude 30 m.

4.7 Vliv vyhlídky na oblast s odlišnou rychlostí volného toku na stav řidiče

Pokud se před řidičem objeví oblast s odlišnou RVT, je třeba, aby přizpůsobil svoji jízdu takovým způsobem, aby do této oblasti vjel maximálně touto rychlostí.

Pokud je hodnota rychlosti větší než RVT v oblasti aktuální pozice vozidla, je třeba zajistit, aby došlo k aktualizaci stavu řidiče v okamžiku kdy se do této oblasti dostane, aby mohl začít zrychlovat na RVT v této oblasti. Vyřešením následující kvadratické rovnice dostaneme dobu, za kterou se tam řidič dostane za předpokladu že nebude měnit stav.

$$\frac{a}{2}t^2 + v_0t - d = 0$$

a ... aktuální zrychlení řidiče
 v_0 ... aktuální rychlost řidiče
 d ... vzdálenost k oblasti s odlišnou RVT

Pokud je výsledna doba kratší než časové kvantum aktuálního stavu řidiče, řidič si zaznamená, že má svůj stav aktualizovat za tuto kratší dobu. V opačném případě je výsledek bezvýznamný.

V situaci kdy je hodnota rychlosti menší než RVT v oblasti aktuální pozice vozidla, je třeba zjistit jak ještě dlouho řidič v aktuálním stavu může setrvat než bude potřeba zpomalovat na tutu rychlost popřípadě jestli už není třeba brzdít. Budeme vycházet z následující rovnice:

$$s_1 + s_2 = d$$

s_1 ... vzdálenost, kterou vozidlo urazí ještě v aktuálním stavu
 s_2 ... vzdálenost, kterou urazí během zpomalování na RVT nové oblasti
 d ... vzdálenost k oblasti s odlišnou RVT

Jednotlivé vzdálenosti můžeme vyjádřit pomocí času, zrychlení a počáteční rychlosti.

$$\frac{a_1}{2}t_1^2 + v_1t_1 + \frac{a_2}{2}t_2^2 + v_2t_2 = d \quad (4.1)$$

a_1 ... aktuální zrychlení
 v_1 ... aktuální rychlost
 a_2 ... zrychlení (< 0), kterým chce řidič zpomalovat na RVT v nové oblasti
 v_2 ... rychlost, kterou řidič dosáhne v okamžiku přepnutí do stavu zpomalování

Přičemž

$$v_2 = v_1 + a_1t_1 \quad (4.2)$$

$$t_2 = \frac{\Delta v}{a_2} = \frac{v_{RVT} - v_2}{a_2} = \frac{v_{RVT} - (v_1 + a_1t_1)}{a_2} \quad (4.3)$$

Po dosazení 4.2 a 4.3 do 4.1 dostaneme

$$\frac{a_1}{2}t_1^2 + v_1t_1 + \frac{(v_{RVT} - (v_1 + a_1t_1))^2}{2a_2} + \frac{(v_1 + a_1t_1)(v_{RVT} - (v_1 + a_1t_1))}{a_2} = d$$

Po úpravě dostáváme kvadratickou rovnici

$$t_1^2(a_1a_2 - a_1^2) + 2t_1(v_1(a_2 - a_1)) + (v_1 - v_{RVT})^2 - 2v_1(v_1 - v_{RVT}) - 2a_2d = 0 \quad (4.4)$$

Výsledek rovnice 4.4 je doba, kterou může ještě řidič v aktuálním stavu strávit než bude potřeba aktualizovat. Pokud je doba kratší než aktuální časové kvantum stavu, řidič si zaznamená, že má svůj stav aktualizovat za tuto kratší dobu. Pokud je výsledek menší nebo roven nule, je třeba se přepnout do stavu brždění. Hodnota zrychlení při brždění bude

$$a_d = \frac{(v_{RVT}^2 - v_1^2)}{2d} \quad (4.5)$$

a doba brždění bude

$$t_d = \frac{v_{RVT} - v_1}{a_d} \quad (4.6)$$

4.8 Vliv vozidla před řidičem na jeho stav

Pokud řidiči v jeho jízdě vadí vozidlo před ním a nemá možnost jej předjet, má za úkol si udržovat za ním bezpečnou vzdálenost popsanou v sekci 4.6. V tomto oddílu probereme jakým způsobem se snaží do bezpečné vzdálenosti dostat v různých situacích kdy vozidlo před ním zrychluje, má konstantní rychlost nebo brzdí.

Ještě než začneme rozebírat jednotlivé případy, je třeba zmínit pomocnou techniku, která se používá při řešení každého případu. Rovnice, která popisuje v závislosti na čase vzdálenost vozidel A a B takových, že A se nachází před B a jedou ve stejném směru, má tvar

$$d(t) = \frac{(a_2 - a_1)}{2}t^2 + (v_2 - v_1)t + d_0 \quad (4.7)$$

a_1 ... zrychlení vozidla B

a_2 ... zrychlení vozidla A

v_1 ... rychlost vozidla B v čase $t = 0$

v_2 ... rychlost vozidla A v čase $t = 0$

d_0 ... vzdálenost mezi vozidly v čase $t = 0$

Důležitou roli hraje okamžik, kdy vzdálenost mezi vozidly je minimální. K výpočtu tohoto časového bodu využijeme derivaci.

$$d'(t) = (a_2 - a_1)t + (v_2 - v_1)$$

Abychom zjistili extrém, položíme derivaci rovnou nule.

$$\begin{aligned} (a_2 - a_1)t + (v_2 - v_1) &= 0 \\ t &= -\frac{v_2 - v_1}{a_2 - a_1} \end{aligned} \quad (4.8)$$

Abychom zjistili, jestli je extrém minimum, najdeme druhou derivaci a najdeme podmínky, v kterých je kladná.

$$\begin{aligned} d''(t) &= a_2 - a_1 \\ a_2 - a_1 &> 0 \\ a_2 &> a_1 \end{aligned}$$

Jelikož se vždy budeme snažit najít zrychlení pro vozidlo B takové, aby nenarazilo do vozidla A, nikdy mu nedáme zrychlení větší než má vozidlo A, tedy v každém případě testování zrychlení pro vozidlo B bude vzdálenost mezi vozidly v čase extrému minimum.

V každé konfiguraci se budeme snažit navrhnout vhodné zrychlení pro vozidlo B tak, aby se rozumně dostalo do bezpečné vzdálenosti za vozidlo A. Pokud navržené zrychlení bude větší než aktuální zrychlení vozidla B, není třeba se jím zabývat a jeho řidič již nebude muset brát ohled na vozidlo A. Pokud nebude uvedeno jinak, časové kvantum stavu s navrženým zrychlením bude vždy tak velké, aby řidič dojel do požadované rychlosti, jejíž hodnota se určí podle chování vozidla A. Pokud zrychluje, požadovaná rychlost bude RVT, pokud má konstantní rychlost, požadovaná rychlost bude rychlost vozidla a pokud brzdí, požadovaná rychlost bude 0 km/h.

Zvolíme minimální vzdálenost za vozidlem A, kterou nesmí vozidlo B při novém zrychlení překročit. Tato vzdálenost se určí jako maximum z hodnot 1.5m a poloviny aktuální vzdálenosti k vozidlu A. Po navržení tohoto zrychlení se musí zkontrolovat, zda se nenaruší zvolená minimální vzdálenost. Pokud ano, je třeba najít takové zrychlení, aby se vozidlo B dostalo k vozidlu A maximálně na tuto vzdálenost. Do rovnice 4.7 dosadíme hodnotu 4.8, položíme ji rovnou minimální vzdálenosti a vyjádříme hodnotu a_1 .

$$a_1 = a_2 - \frac{v_2 - v_1}{2(d_0 - d_{min})}$$

d_{min} ... minimální vzdálenost mezi vozidly A a B

Časové kvantum stavu nastavíme tak, aby k aktualizaci došlo v okamžiku, kdy se vozidlo dostane do minimální vzdálenosti. Pokud nedojde k narušení minimální vzdálenosti, vozidlo B si zvolí navržené zrychlení, které popíšeme v následujících třech logických odstavcích.

„Vozidlo přede mnou zrychluje“

V případě že vozidlo před řidičem zrychluje, rozčleníme situaci do dvou situací.

Pokud je vzdálenost mezi vozidly větší než bezpečná vzdálenost podle vozidla B v aktuální rychlosti vozidla A a rychlost vozidla B je větší než vozidla A, zkusíme brzdít tak, aby se vozidlo B nedostalo k vozidlu A blíže než na bezpečnou vzdálenost. Budeme vycházet z následující rovnice:

$$\frac{(a_2 - a_1)}{2}t^2 + (v_2 - v_1)t + d_0 = c_t(v_2 + a_2t) \quad (4.9)$$

c_t ... konstanta doby bezpečné vzdálenosti vozidla B

Všimněme si, že pravá strana rovnice 4.9 reprezentuje bezpečnou vzdálenost za vozidlem A v závislosti na rostoucí rychlosti vozidla A.

Do 4.9 dosadíme 4.8 a vyjádříme a_1 .

$$a_1 = \frac{-(v_2 - v_1)^2 + 2c_t a_2 (v_2 - v_1) + 2a_2 (d - c_t v_2)}{2(d - c_t v_2)}$$

Pokud by výsledné brždění bylo příliš veliké, za a_1 zvolíme standardní brždění z aktuální rychlosti B na rychlost 0 km/h a provedeme kontrolu narušení minimální vzdálenosti popsané na začátku tohoto oddílu.

Nyní se podíváme na případ, kdy se vozidlo B nachází v bližší vzdálenosti než bezpečné podle aktuální rychlosti vozidla A nebo rychlost vozidla B je menší než vozidla A. Budeme vycházet z následující rovnice:

$$\frac{(a_2 - a_1)}{2}t^2 + (v_2 - v_1)t + d_0 = c_t v_{RVT} \quad (4.10)$$

c_t ... konstanta doby bezpečné vzdálenosti vozidla B

v_{RVT} ... aktuální rychlost volného toku vozidla B

Opět pravá strana rovnice určuje bezpečnou vzdálenost. Tentokrát se předpokládá, že vozidlo A bude zrychlovat do RVT vozidla B v jeho aktuální pozici. Cílem je získat takové zrychlení, které nám zajistí, že vozidlo B dosáhne rychlosti v_{RVT} v okamžiku kdy se dostane do bezpečné vzdálenosti za vozidlem A v této rychlosti.

$$\begin{aligned} v_{RVT} &= v_1 + a_1 t \\ t &= \frac{v_{RVT} - v_1}{a_1} \end{aligned} \quad (4.11)$$

Dosazením 4.11 do 4.10 a následnou úpravou dostaneme kvadratickou rovnici:

$$2a_1^2(d_0 - c_t v_{RVT}) + a_1(v_{RVT} - v_1)(2(v_2 - v_1) - (v_{RVT} - v_1)) + a_2(v_{RVT} - v_1)^2 = 0 \quad (4.12)$$

Výsledné zrychlení a_1 vezmeme jako minimum z výsledku rovnice 4.12 a standardního zrychlení z aktuální rychlosti vozidla B na RVT jeho aktuální pozice, aby vozidlo zbytečně zběsile nezrychlovalo jenom aby se dostalo včas do bezpečné vzdálenosti vozidla. Pak již jen zbývá zkontrolovat, zda se při tomto zrychlení nepřekročí minimální vzdálenost.

„Vozidlo přede mnou má konstantní rychlost“

Nejprve se zkontroluje, zda není vzdálenost mezi vozidly rovna bezpečné vzdálenosti v rychlosti vozidla A podle řidiče vozidla B. Pokud ano a rozdíl rychlostí není větší než 5 km/h, řidič vozidla B si nastaví svoji rychlost na hodnotu rychlosti vozidla A, čímž si zajistí, že vzdálenost mezi nimi se již nebude měnit do té doby než jeden z nich se rozhodne brzdit nebo zrychlovat.

V opačném případě je vzdálenost menší nebo větší než bezpečná vzdálenost podle řidiče vozidla B.

Pokud je vzdálenost menší, úkolem řidiče vozidla B je dostat se do bezpečné vzdálenosti. Pokud je rychlost vozidla B menší než rychlost vozidla A, k řešení použijeme rovnici 4.12 s tím rozdílem, že hodnota $a_2 = 0$ a $v_{RVT} = v_2$. Po úpravě dostaneme výsledné zrychlení

$$a_1 = -\frac{(v_2 - v_1)^2}{2(d - c_t v_2)}$$

V případě, že rozdíl rychlostí vozidel je v toleranci 5 km/h, bude potřeba na chvíli zpomalit a pak opět zrychlit tak, aby se vzdálenost vozidel dostala na hodnotu bezpečné vzdálenosti. Budeme vycházet z následující rovnice:

$$s_A - s_{B1} - s_{B2} + d_0 = d_b$$

s_A ... vzdálenost, kterou urazí vozidlo A během manévru vozidla B

s_{B1} ... vzdálenost, kterou vozidlo B urazí při zpomalení

s_{B2} ... vzdálenost, kterou vozidlo B urazí při následném zrychlení

d_0 ... vzdálenost mezi vozidly na začátku manévru

d_b ... velikost bezpečné vzdálenosti

Vzdálenosti lze vyjádřit pomocí počáteční rychlosti, zrychlení a času:

$$v_A t_1 + v_A t_2 - \left(\frac{a_{B1}}{2} t_1^2 + v_{B1} t_1\right) - \left(\frac{a_{B2}}{2} t_2^2 + v_{B2} t_2\right) + d_0 = d_b$$

v_A ... rychlost vozidla A v první a druhé fázi manévru

a_{B1} ... zrychlení vozidla B v první fázi manévru

v_{B1} ... rychlost vozidla B na začátku první fáze manévru

a_{B2} ... zrychlení vozidla B v druhé fázi manévru

v_{B2} ... rychlost vozidla B na začátku druhé fáze manévru

t_1 ... doba trvání prvního manévru

t_2 ... doba trvání druhého manévru

Zrychlení a_{B1} , resp. a_{B2} zvolíme jako standardní zrychlení z rychlosti 10 km/h na 0 km/h, resp. z rychlosti 0 km/h na 10 km/h. Nakonec si zvolíme podmínku, že $t_1 = t_2$, čili doba trvání první fáze manévru bude stejně dlouhá jako druhá fáze manévru. Po dosazení dostáváme kvadratickou rovnici:

$$t^2 \left(-\frac{3a_{B1}}{2} - \frac{a_{B1}}{2}\right) + 2t(v_{A1} - v_1) + d_0 - d_b = 0 \quad (4.13)$$

Výsledkem rovnice 4.13 je časové kvantum, které má řidič strávit ve stavu brždění zrychlením a_{B1} .

Nakonec pokud je rychlost vozidla A menší než vozidla B, vozidlo B si zvolí brždění o standardním zrychlení z rychlosti 10 km/h na rychlost 0 km/h a další analýzu ponechá na algoritmu, který kontroluje narušení minimální vzdálenosti, popsáno na začátku tohoto oddílu.

Pokud je vzdálenost mezi vozidly A a B větší než bezpečná vzdálenost podle řidiče vozidla B, je třeba najít časové kvantum, během kterého může řidič v aktuálním stavu ještě zůstat, než se bude muset přepnout do stavu zpomalování tak, aby se dostal do bezpečné vzdálenosti za vozidlo A, pokud je rychlost vozidla A menší než rychlost volného toku podle řidiče B v jeho aktuální pozici. Budeme vycházet z následující rovnice:

$$s_A - s_{B1} - s_{B2} + d_0 = d_b$$

s_A ... vzdálenost, kterou urazí vozidlo A během manévru vozidla B

s_{B1} ... vzdálenost, kterou vozidlo B urazí ještě v aktuálním stavu

s_{B2} ... vzdálenost, kterou vozidlo B urazí ve fázi zpomalování

d_0 ... vzdálenost mezi vozidly na začátku manévru

d_b ... velikost bezpečné vzdálenosti

Vzdálenosti lze opět vyjádřit pomocí počáteční rychlosti, zrychlení a času:

$$v_A(t_1 + t_2) + \frac{a_A}{2}(t_1 + t_2)^2 - \left(\frac{a_{B1}}{2} t_1^2 + v_{B1} t_1\right) - \left(\frac{a_{B2}}{2} t_2^2 + v_{B2} t_2\right) + d_0 = d_b \quad (4.14)$$

v_A ... rychlost vozidla A v první a druhé fázi manévru
 a_A ... zrychlení vozidla A v první a druhé fázi manévru
 a_{B1} ... zrychlení vozidla B v první fázi manévru
 v_{B1} ... rychlost vozidla B na začátku první fáze manévru
 a_{B2} ... zrychlení vozidla B při zpomalování
 v_{B2} ... rychlost vozidla B na začátku fáze zpomalování
 t_1 ... doba trvání první fáze manévru
 t_2 ... doba trvání zpomalení

Přičemž platí

$$\begin{aligned}
 v_{B2} &= v_{B1} + a_{B1}t_1 \\
 t_2 &= \frac{v_{Bf2} - v_{B2}}{a_{B2}}
 \end{aligned} \tag{4.15}$$

v_{Bf2} ... rychlost vozidla B po fázi zpomalování

Dosazením 4.15 do 4.14 a následnou úpravou dostaneme kvadratickou rovnici:

$$\begin{aligned}
 &t_1^2(a_{B1}^2(a_A - a_{B2}) + (a_A - a_{B1})(a_{B2}^2 - 2a_{B1}a_{B2})) + \\
 &2t_1((v_a - v_{B1})(a_{B2}^2 - a_{B1}a_{B2}) + \Delta v a_A(a_{B2} - a_{B1})) + \\
 &2a_{B2}\Delta v(v_A - v_{B1}) + \Delta v^2(a_A - a_{B2}) - 2a_{B2}^2(d_b - d_0) = 0
 \end{aligned} \tag{4.16}$$

$$\Delta v = v_{Bf2} - v_{B1}$$

Výsledek rovnice 4.16 je časové kvantum, které může řidič vozidla B ještě strávit v aktuálním stavu než ho bude muset aktualizovat. Pokud výsledek není kladný, je třeba začít brzdit okamžitě.

„Vozidlo přede mnou brzdí“

V případě, že vozidlo před řidičem brzdí, řidič předpokládá, že vozidlo bude brzdit až do zastavení. Předpočítá si vzdálenost, za kterou řidič zastaví a podle toho bude vědět vzdálenost, kterou má sám k dispozici k zastavení za tímto vozidlem. Zrychlení, kterým bude zpomalovat až do fáze zastavení, se určí jako minimum z hodnot standardního zpomalení z aktuální rychlosti na 0 km/h a zrychlení určeného vzorečkem 4.5 pro $v_{RVT} = 0$. Zvolené zrychlení podrobíme testu o narušení minimální vzdálenosti. Pokud minimální vzdálenost nebude narušena, najdeme časové kvantum, které může řidič ve svém stavu ještě zůstat, než bude muset brzdit velikostí zvoleného zrychlení. K tomuto účelu použijeme rovnici 4.16 s tím, že $a_A = 0$, $v_A = 0$ a k vzdálenosti mezi vozidly se přičte vzdálenost, kterou vozidlo A ujede než zastaví.

4.9 Predikční model v křižovatce

K tomu aby byl řidič schopný projet křižovatkou, potřebuje si vytvořit představu o tom jak skrze ní budou projíždět ostatní vozidla, kterým má buď dát přednost, nebo se nachází před ním a řidič potřebuje vědět jak moc ho můžou při průjezdu křižovatkou zdržovat.

K této predikci se využívá technika diskrétní simulace. Díky časovým kvantům, které vymezují dobu trvání stavu řidiče se může sestavit časový kalendář, kde jednotlivé položky budou vymezovat okamžiky změny simulované soustavy objektů. Tato soustava objektů může být množina po sobě jdoucích vozidel obsažených v dráze jednoho vozidla. Doba do nejbližší změny soustavy se určí jako minimum množiny časových kvant, které zbývají do aktualizace stavů jednotlivých řidičů zahrnutých v diskrétní simulaci. Řidiči si po uplynutí tohoto minima aktualizují pozici svého vozidla a popřípadě i svůj stav, pokud došlo k vyčerpání jeho časového kvanta. Tímto způsobem si řidič dokáže odsimulovat sám sebe jak bude schopný křižovatkou projet a stejně i vozidla obsažená v jízdnicích pruzích, jejichž dráhu bude při průjezdu křižovatkou křížit.

4.10 Zastavit před nebo v křižovatce?

V tomto oddílu budeme označovat pojem hranice křižovatky z pohledu řidiče jako hranici prvního průsečíku jeho jízdnicího pruhu v křižovatce s ostatními pruhy křižovatky.

Pokud se řidič nachází před křižovatkou a usoudí, že křižovatkou nebude schopný projet, bude muset zastavit a počkat, až bude volno. V průběhu dojíždění na místo zastavení si řidič v každém kroku simulace kontroluje, zda se situace v křižovatce nezměnila tak, že by jí mohl projet. Jsou dvě možnosti jak si zvolit místo zastavení, které závisí na tom, kvůli kterému vozidlu nemůže řidič křižovatkou projet. Pokud vozidlo má přednost v jízdě podle nastavení křižovatky, řidič si zvolí místo zastavení 0.5m před hranicí křižovatky. Pokud vozidlu nemusí dávat přednost, ale vozidlo se už nachází za hranicí křižovatky, místo zastavení se nachází 0.5m před hranicí průsečíku jízdnicího pruhu řidiče s jízdnicí pruhem, v kterém se toto vozidlo nachází. Jakmile se řidič dostane za hranici křižovatky a nikomu v této křižovatce nedává přednost, již nekontroluje, zda nemá někomu dát přednost v jízdě. Nyní si musí ostatní řidiči kontrolovat, zda jeho vozidlo nebude bránit jejich průjezdu křižovatkou.

Pokud vozidlo dojíždí do místa v křižovatce s tím, že někomu dává přednost, je třeba tomuto vozidlu uvolnit cestu, aby mohlo oblast křižovatky co nejdříve uvolnit. Z tohoto důvodu řidiči, kteří mají povinnost tomuto vozidlu dát

přednost, zastaví svá vozidla před hranicí křižovatky i v případě, že by při jejich průjezdu křižovatkou nedošlo ke srážce. Řidiči, kteří nemají povinnost tomuto vozidlu dát přednost, zastaví 0.5m před hranicí průsečíku jejich jízdního pruhu s jízdním pruhem, v kterém se toto vozidlo nachází. Takto bude mít toto vozidlo po tom, co dá přednost vozidlu, kvůli kterému v křižovatce zastavilo, nachystanou volnou cestu k vyjetí z křižovatky.

4.11 Impulsy k změně jízdního pruhu

Řidič má možnost se přeřadit do vedlejších pruhů pod daným úhlem. Vedou ho k tomu okolnosti v silničním provozu, mezi které patří směry jízdních pruhů před křižovatkou, zúžení vozovky, chování vozidla před řidičem a volný pravý jízdní pruh. V následujících bodech rozebereme jednotlivé důvody k přeřazení seřazené v pořadí, v kterém je řidič kontroluje.

- Jako první řidič kontroluje zda se nachází před křižovatkou a jestli se nachází v jízdním pruhu, z kterého lze na křižovatce jet jeho požadovaným směrem. Pokud ne, je třeba se přeřadit do nejbližšího jízdního pruhu, z kterého lze tímto směrem jet. Podle počtu jízdních pruhů, přes které se musí přeřadit, se odvodí vzdálenost před křižovatkou, kterou nesmí řidič překročit, aby se mohl pohodlně přeřadit do správného pruhu.
- Jako druhé v pořadí řidič kontroluje zda se vozovka nezužuje a jestli není v pruhu, který končí. Pokud ano, opět podle počtu jízdních pruhů, přes které se musí přeřadit, aby mohl pokračovat dál v jízdě, se odvodí vzdálenost před hranicí zúžení vozovky, kterou nesmí řidič překročit, aby se mohl pohodlně přeřadit do správného pruhu.
- Třetí v pořadí se řidič dívá na vozidlo před ním. Pokud má o 5 km/h menší rychlost nebo pokud by bez něj řidič na základě kontextu jeho vozidla zrychlil, bude se snažit o jeho předjetí. Pokud se řidič nachází před křižovatkou, z jízdního pruhu, přes který ho chce předjet, musí také být možnost jet směrem, kterým chce řidič v této křižovatce jet.
- Jako poslední v pořadí si řidič kontroluje, jestli není po jeho pravici volný pruh, ve kterém by mohl strávit dostatečně dlouhou dobu, pokud dochází později k zúžení vozovky. Pokud pruh vyhovuje požadavkům řidiče, přeřadí se do něho.

4.12 Aktualizace stavu

K aktualizaci stavu vede několik situací. Zejména k nim patří situace kdy vyprší jeho časové kvantum. Dále pak události v silničním provozu, které

jeho změnu vyžadují. Mezi ně patří hlavně události týkající se stavu vozidla před řidičem. Provádí se také v případě kdy řidič stojí před křižovatkou, dává přednost a potřebuje si zjistit, jestli by byl schopný křižovatkou projet, kdyby v daný okamžik svůj stav aktualizoval.

Na začátku aktualizace se porovná aktuální rychlost vozidla a rychlost volného toku oblasti, v které se vozidlo řidiče nachází. Pokud je větší, řidič se rozhodne zrychlit na tuto rychlost standardním zrychlením popsáném v sekci 4.5 a velikost časového kvanta bude taková aby po jeho vypršení řidič dosáhl rychlosti volného toku. Poté se provede kontrola oblastí s odlišnou rychlostí volného toku, které se nachází před řidičem, jejíž postup je popsán v sekci 4.7. Po ní následuje kontrola vozidla před řidičem popsána v sekci 4.8. V neposlední řadě se zkontroluje zda není třeba změnit jízdní pruh v pořadí popsáném v sekci 4.11 a nakonec se zkontrolují vozidla napravo a nalevo od řidiče, jestli se nechtějí přeradit před řidiče a jestli není možné jim to umožnit.

4.13 Další implementace

Kromě výše popsáného modulu chování "driver1" jsou v projektu k dispozici další dva. Liší se mezi sebou koeficientem překročení maximální povolené rychlosti a konstantou určující bezpečnou vzdálenost za vozidlem.

Další implementace by bylo možné odvodit z implementace "driver1" pomocí mechanismu dědičnosti a předefinovat některé z klíčových funkcí. Mezi ně například patří funkce určující, zda je či není možné zařadit se před/za vozidlo v sousedním pruhu na základě vzdálenosti, rychlosti a zrychlení onoho vozidla. Mezi zajímavé možnosti by určitě patřila změna standardního zrychlení/zpomalení z dané počáteční rychlosti na danou koncovou rychlost v závislosti na náladě řidiče spolu s aktuální dopravní situací.

Kapitola 5

Shrnutí

5.1 Výsledná podoba projektu

Výsledkem projektu je grafické uživatelské rozhraní s nabídkou komponent silniční sítě, které jsou k dispozici ve formě ikoněk. Uživatel si může zvolit jednu z komponent a umístit ji na interaktivní mapu.

K dispozici je část silničního koridoru ve formě zatáčky doleva, doprava nebo rovného úseku. Tyto koridory je možné na sebe navazovat a měnit v nich počet jízdních pruhů. Uživatel má k dispozici algoritmus propojení dvou koridorů, jehož princip je popsán v sekci 2.6. Ke každému koridoru jsou během simulace k dispozici statistická data hovořící o propustnosti úseku. Popis jednotlivých dat je v sekci 3.8. Pokud koridor nenavazuje na jiný koridor, uživatel má možnost k němu dodat informace ohledně očekávané hustoty vozidel, které do něj budou vjíždět. Další položka k dispozici je křižovatka, jejíž tvar je pevně dán a je možné v ní upravit přednost v jízdě. Vstupní vozovky jsou na sebe kolmé. Lze je však rozšiřovat a určovat kterými směry se z nového jízdního pruhu v křižovatce bude možno vydat. Ke každé vstupní vozovce lze nastavit podíl toku provozu v jednotlivých směrech do křižovatky.

5.2 Test simulátoru

V Olomouci byla nedávno uzavírka hlavního tahu z Mohelnice na Brno a doprava se musela odklonit na užší silnice v oblasti okrajové části města. Na této trase se nacházela křižovatka, v níž jedna z vozovek, po které se tento tranzit ubíral, nebyla hlavní. Silniční obsluha musela zvážit, jestli nebude třeba upravit přednost v jízdě v této křižovatce, aby se netvořily kolony.

Naším simulátorem se pokusíme celou situaci odsimulovat a najít k ní řešení. V příloženém CD v adresáři *samples* se nachází tři soubory popisující tuto situaci. První se jménem *olomouc0.sim* vykresluje situaci před zahájením

uzavírky, soubor *olomouc1.sim* pak po zahájení uzavírky s ponecháním původní úpravy přednosti v jízdě a nakonec soubor *olomouc2.sim* mapuje situaci při uzavírce s tím, že hlavní silnice v křižovatce vede po trase, kudy jezdí vozidla, která měla jet po hlavním tahu. Po odsimulování modelu *olomouc0.sim* jde vidět, že křižovatka nemá žádné problémy provoz zvládnout. Po zahájení uzavírky je patrné, že hlavní proud vozidel, který nejede po hlavní silnici, musí ve spoustě případech stát kvůli pár vozidlům, které se ubírají po hlavní silnici a tím se celá kolona zdržuje zbytečným brzděním a zrychlováním. Na této vedlejší vozovce byla před uzavírkou odsimulována průměrná rychlost něco přes 40 km/h. Nyní tato rychlost klesla na zhruba 10 km/h. V modelu *olomouc2.sim* se tato hodnota pohybuje na rychlosti bezmála 25 km/h, což je znatelný rozdíl. Horší situace ale nastává na vozovce, která byla původně hlavní a nyní je vedlejší. Tvoří se zde kolony. Ani jedno z řešení není ideální, ale vzhledem k tomu, že po této vozovce jezdí v drtivé většině obyvatelé místní části Olomouce, o uzavírce se brzy dozví a svoji cestu naplánují tak, aby nevedla přes tuto křižovatku. Řešením situace je tedy změnit úpravu přednosti v jízdě a informovat o této situaci obyvatelé místní části města.

5.3 Jak projekt rozšířit

Projekt lze rozšířit hned v několika oblastech.

Mezi první patří dopravní informace. Do jízdních pruhů lze vložit jakýkoli druh informace mezi které může patřit maximální povolená rychlost, informace o velikosti stoupání/klesání vozovky, zákaz předjíždění, atd..

Silniční koridory vedou buď do křižovatky, nebo ven z modelu a nedá se z nich nikde odbočit. Jejich rozšíření by spočívalo v možnosti jejich rozštěpení do nových užších jízdních koridorů, nebo sloučení několika jízdních koridorů do jednoho. Díky tomu by se dali například vytvořit odbočovací jízdní pruhy jako na dálnici, rozšíření vozovky směrem doleva a jiné. Zároveň by se k tomu musely dodat informace ohledně podílu toku provozu do jednotlivých větví vycházejících z koridoru a tím pádem i počítat s takto navrženou vozovkou u implementace řídičů.

U rozšíření křižovatek přichází v úvahu v první řadě její tvar, tzn. jaký úhel mezi sebou svírají jednotlivé vstupní vozovky. S pomocí algoritmu propojení jízdních pruhů popsaném v sekci 2.6 a nalezení průsečíků dvou jízdních pruhů popsaném v sekci 3.4, lze poměrně jednoduše vstupní vozovky propojit a nalézt průsečíky propojovacích pruhů v křižovatce. Takové rozšíření by znamenalo hlavně zásah do grafického rozhraní uživatele. Z pohledu uložení modelu do souboru nebo dopravních informací uložených v jízdních pruzích by se nic nezměnilo. V druhé řadě rozšíření by u křižovatek figurovaly se-

mafony. Znamenalo by to vytvořit cyklicky se měnící mechanismus, který by přijímal informaci o tom, že proběhl krok simulace, aby mohl aktualizovat svůj stav. Zároveň by bylo třeba navrhnout komunikační rozhraní mezi řidičem a semaforem a při implementaci řidiče na semafony myslet.

Díky rozhraní řidiče popsaném v sekci 3.6 a dostupným informacím z jeho kontextu je k dispozici široká škála možností, jak umělou inteligenci řidiče naimplementovat. Např. podle [3] by bylo možné myslet na maximální kapacitu lidského mozku s ohledem na počet paralelně pozorovaných objektů, jeho únavu a jiné aspekty ovlivňující jeho schopnost vnímat. Modul řidiče popsaný v kapitole 4 by bylo možné rozšířit na schopnost předjíždět vozidla s použitím jízdního pruhu, kde se vozidla ubírají opačným směrem. Díky schopnosti diskrétní simulace, kterou používají řidiči v křižovatce, by byl schopný odsimulovat řidiče jedoucí v protisměru a podle toho se rozhodnout, jestli by byl schopný předjet nebo ne.

Kapitola 6

Srovnání s projektem Simulace automobilového provozu

Projekt Simulace automobilového provozu [5] je zaměřený na širší použití simulace. Je zde možné sbírat podrobnější data týkající se jednotlivých vozidel a jejich řidičů, zejména pak jejich úspěšnosti průjezdu silniční sítí. Je zde možnost vytyčit trasu, kudy řidič má jet a testovat jednotlivé implementace řidičů jak budou schopni tuto trasu zvládnout. Měří se doba jejich jízdy, průměrná rychlost a spotřeba paliva. Mezi další výhody patří možnost vytvořit křižovatku, do které vede n silničních koridorů. Ke každé křižovatce je ale nutné dodat soustavu semaforů, což na jednu stranu je výhodné, že je možné testovat seřízení semaforů na křižovatce, ale není možné testovat schopnost křižovatky být bez semaforů.

Mezi horší stránky projektu patří reprezentace silniční sítě. Je reprezentovaná jako orientovaný graf, kde uzly představují křižovatky a hrany silniční koridory. Nelze zde věrohodně namodelovat zatáčky o daném poloměru, protože mezi křižovatkami vedou pouze rovné silniční koridory. Jediná možnost jak namodelovat efekt zatáčky je zadat maximální povolenou rychlost v dané oblasti, což stírá možnost modelovat pohyb vozidel s rozdílnými fyzikálními vlastnostmi. Mezi další nevýhody patří reprezentace křižovatky jako jeden bod, tedy křižovatka nemá žádné rozměry. Pokud má dát řidič přednost jinému vozidlu, je třeba, aby pouze kontroloval, kdy oblast jeho vozidla bude projíždět bodem křižovatky. Jelikož je síť reprezentovaná jako graf, nejsou zde žádné vstupní a výstupní silniční koridory. Vozidla se generují přímo doprostřed silničního koridoru. Lze tedy stěží modelovat očekávanou hustotu dopravy na jednotlivých silničních koridorech.

Popisovaný projekt není absolutní mikrosimulace. Je zde "vyšší moc", která kontroluje rozhodnutí každého řidiče v každém kroku simulace. Pokud chování řidiče nevyhovuje stanoveným podmínkám, tato moc zakáže řidiči provést požadovanou akci, čili řidič si nemá možnost vytvořit libovolnou strategii jízdy, protože nikdy neví, jestli mu požadovaná akce bude dovolena.

Kapitola 7

Závěr

V předloženém projektu je představen nástroj pro mikrosimulaci dopravy. Podařilo se navrhnout a popsat reprezentaci silniční sítě, její strukturu a provázanost spolu s možnostmi rozšíření tohoto návrhu.

Jeho současná podoba dovoluje simulovat a řešit několik reálných situací. Při návrhu křižovatky simulátor dokáže odhadnout, jestli úprava přednosti v jízdě a parametry vstupních vozovek budou vyhovovat očekávané hustotě dopravy. Mezi parametry vozovek patří jejich šířka a směry jednotlivých jízdnicích pruhů, kterými z nich lze v křižovatce jet. Mimo oblast křižovatky lze simulátor použít například v situacích, kdy se plánují práce na víceproudové vozovce. Pokud se budou muset uzavřít některé jízdnicí pruhy, je potřeba zjistit, jestli zbývající jízdnicí pruhy budou schopny provoz odbavit, nebo jestli bude třeba část provozu odklonit jinými prostředky.

Implementací popsané umělé inteligence řidiče se podařilo přednést možný přístup k řešení mikrosimulace dopravy se snahou omezit nároky na výpočetní sílu procesoru díky tomu, že řidiči se přepínají mezi stavy, které mají danou dobu trvání, a nemusí v každém kroku simulace kontrolovat stav svého okolí.

Literatura

- [1] Akcelik R., Besley M.: *Acceleration and deceleration models*. 23rd Conference of Australian Institutes of Transport Research (CAITR 2001), Monash University, Melbourne, Australia
- [2] Bednařík M., Šíroková M.: *Rychlost a bezpečná jízda*. SPN, Praha 1972
- [3] Brandejský T.: *Mentální modely řidičů uvažující únavu a ostatní limity procesní kapacity*, ČVUT, Praha 2003
- [4] Chui Samantha, Hsiang T.J., Shen Jennifer: *S.I.M. Traffic - Project Design Document*, Stanford University, Silicon Valley.
- [5] Gregor I.: *Simulace automobilového provozu*, bakalářský projekt MFF UK, Praha 2006
- [6] Sanford, K. L.: *A Complex System Approach to Traffic Simulation* Masters Thesis, Carnegie Mellon University Pittsburgh, PA, 1994
- [7] *Základní parametry výhledové silniční sítě kraje*. ÚPP-ÚG dopravy, silnic II. a III. třídy na území Olomouckého kraje.

Příloha

Obsah přiloženého CD

Součástí bakalářské práce je přiložené CD obsahující popisovaný projekt. Kořenový adresář disku obsahuje následující adresáře a soubory:

- Soubor *simulace.jar* obsahuje aplikaci, k jejíž spuštění je třeba Java Runtime Environment 5.0.
- Soubor *manual.pdf* obsahuje uživatelskou příručku k ovládání grafického rozhraní programu.
- Soubor *bakalarska-prace.pdf* je kopii této práce.
- Soubor *build.xml* obsahuje ant-skript, pomocí kterého lze přeložit zdrojové soubory programu obsažené v adresáři *source*. K tomuto přeložení je třeba mít k dispozici JDK 5.0.
- Adresář *dokumentace* obsahuje programátorskou dokumentaci aplikace.
- Adresář *samples* obsahuje příklady simulačních modelů.
- Adresář *source* obsahuje zdrojové soubory aplikace.